What is claimed:

1. A computer program product embodied on one or more computer-readable media, the computer program product adapted for efficiently transforming extensible structured documents and comprising:

computer-readable program code means for identifying a source document type;

computer-readable program code means for specifying one or more fast transformations to be performed on documents of the source document type;

computer-readable program code means for specifying a source node description and a target node description for each of the specified fast transformations;

computer-readable program code means for storing transformation information for each of the specified fast transformations, the transformation information comprising a transformation identifier, the source node description, and the target node description; and

computer-readable program code means for processing incoming source documents to generate output documents using the stored transformation information, further comprising:

computer-readable program code means for receiving a source document;

computer-readable program code means for selecting, manually or based upon a comparison of the received source document to the stored transformation information, zero or more fast transformations to be performed;

computer-readable program code means for applying the selected fast transformations; and

computer-readable program code means for generating one or more output documents using a result of the computer-readable program code means for applying.

2. The computer program product according to Claim 1, wherein the received source document is an Extensible Markup Language (XML) document.

3. The computer program product according to Claim 2, further comprising computer-readable program code means for parsing the XML document.

4. The computer program product according to Claim 1, wherein the received source document is an array-based representation of an Extensible Markup Language (XML) document.

5. The computer program product according to Claim 4, and wherein the computer-readable program code means for applying the selected transformations further comprises computer-readable program code means for manipulating selected nodes by manipulating the array-based representation.

6. The computer program product according to Claim 1, wherein the received source document is a machine-oriented markup language document.

7. The computer program product according to Claim 1, wherein the received source document is an array-based representation of a machine-oriented markup language document.

8. The computer program product according to Claim 1, wherein the received source

document is a parsed representation of an extensible document.

9. The computer program product according to Claim 1, wherein the source node description identifies one or more source nodes in an input document of the source document type and wherein the target node description identifies zero or more target nodes in an output tree to be generated in the one or more output documents.

10. The computer program product according to Claim 1, wherein the general purpose transformation engine is a stylesheet engine.

11. The computer program product according to Claim 10, wherein the stylesheet engine is an Extensible Stylesheet Language (XSL) engine.

12. A system for efficiently transforming extensible structured documents, comprising:

means for specifying fast transformations to be applied to incoming source documents;

means for applying the fast transformations to particular incoming source documents matching criteria of the specified fast transformations; and

means for applying general purpose transformations to incoming source documents not matching criteria of the specified fast transformations.

13. The system according to Claim 12, wherein the means for specifying fast transformations further comprises:

means for identifying a source document type;

means for specifying one or more fast transformations to be performed on documents of the source document type;

means for specifying a source node description and a target node description for each of the specified fast transformations; and

means for storing transformation information for each of the specified fast transformations, the transformation information comprising a transformation identifier, the source node description, and the target node description.

14. The system according to Claim 13, wherein the means for applying the fast transformations further comprises:

means for receiving a source document;

means for selecting, manually or based upon a comparison of the received source document to the stored transformation information, zero or more fast transformations to be performed; and

means for applying the selected fast transformations by manipulating selected nodes of the received source document according to the selected fast transformations.

15. The system according to Claim 12, wherein the received source document is an Extensible Markup Language (XML) document.

16. The system according to Claim 15, further comprising means for parsing the XML

document.

17. The system according to Claim 12, wherein the received source document is an array-based representation of an Extensible Markup Language (XML) document.

18. The system according to Claim 14, wherein the received source document is an array-based representation of an Extensible Markup Language (XML) document, and wherein the means for applying the selected fast transformations by manipulating selected nodes further comprises means for manipulating the array-based representation.

19. The system according to Claim 12, wherein the received source document is a machine-oriented markup language document.

20. The system according to Claim 12, wherein the received source document is an array-based representation of a machine-oriented markup language document.

21. The system according to Claim 12, wherein the received source document is a parsed representation of an extensible document.

22. The system according to Claim 12, wherein the source node description identifies one or more source nodes in an input document of the source document type and wherein the target node description identifies zero or more target nodes in an output tree to be generated in the one

or more output documents.

23. The system according to Claim 12, wherein the general purpose transformation engine is a stylesheet engine.

24. The system according to Claim 23, wherein the stylesheet engine is an Extensible Stylesheet Language (XSL) engine.

25. A method for efficiently transforming extensible structured documents, comprising the steps of:

specifying fast transformations to be applied to incoming source documents;

applying the fast transformations to particular incoming source documents matching criteria of the specified fast transformations; and

applying general purpose transformations to incoming source documents not matching criteria of the specified fast transformations.

26. The method according to Claim 25, wherein the step of specifying fast transformations further comprises the steps of:

identifying a source document type;

specifying one or more fast transformations to be performed on documents of the source document type;

specifying a source node description and a target node description for each of the specified

fast transformations; and

storing transformation information for each of the specified fast transformations, the transformation information comprising a transformation identifier, the source node description, and the target node description.

27. The method according to Claim 26, wherein the step of applying the fast transformations further comprises the steps of:

receiving a source document;

selecting, manually or based upon a comparison of the received source document to the stored transformation information, zero or more fast transformations to be performed; and

applying the selected fast transformations by manipulating selected nodes of the received source document according to the selected fast transformations.

28. The method according to Claim 25, wherein the received source document is an Extensible Markup Language (XML) document.

29. The method according to Claim 28, further comprising the step of parsing the XML document.

30. The method according to Claim 25, wherein the received source document is an array-based representation of an Extensible Markup Language (XML) document.

31. The method according to Claim 27, wherein the received source document is an array-based representation of an Extensible Markup Language (XML) document, and wherein the step of applying the selected fast transformations by manipulating selected nodes further comprises the step of manipulating the array-based representation.

32. The method according to Claim 25, wherein the received source document is a machine-oriented markup language document.

33. The method according to Claim 25, wherein the received source document is an array-based representation of a machine-oriented markup language document.

34. The method according to Claim 25, wherein the received source document is a parsed representation of an extensible document.

35. The method according to Claim 25, wherein the source node description identifies one or more source nodes in an input document of the source document type and wherein the target node description identifies zero or more target nodes in an output tree to be generated in the one or more output documents.

36. The method according to Claim 25, wherein the general purpose transformation engine is a stylesheet engine.

37. The method according to Claim 36, wherein the stylesheet engine is an Extensible Stylesheet Language (XSL) engine.

38. The method according to Claim 27, further comprising the step of preloading one or more templates prior to operation of the step of applying the selected fast transformations.

39. The method according to Claim 25, further comprising using a result of the step of applying the fast transformations and a result of the step of applying general purpose transformations to create an output document, and wherein the source document and/or the output document may be represented as in-memory structures which may have been produced by or may be sent to another software process.